ELSEVIER

Discrete Applied Mathematics 83 (1998) 179–206

DISCRETE APPLIED MATHEMATICS

The complexity of broadcasting in planar and decomposable graphs

Andreas Jakoby[1], Rüdiger Reischuk*, Christian Schindelhauer

Med. Universität zu Lübeck, Institut für Theoretische Informatik, Wallstraße 40, 23560 Lübeck, Germany

Received 30 June 1995; received in revised form 1 April 1996; accepted 28 April 1997

Abstract

Broadcasting in processor networks means disseminating a single piece of information, which is originally known only at some nodes, to all members of the network. The goal is to inform everybody using as few rounds as possible, that is minimize the broadcast time.

Given a graph and a subset of nodes, the sources, the problem to determine its specific broadcast time, or more generally to find a broadcast schedule of minimal length has been shown to be $\mathcal{NP}$-hard. In contrast to other optimization problems for graphs, like vertex cover or traveling salesman, little was known about restricted graph classes for which polynomial time algorithms exist, for example for graphs of bounded treewidth. The broadcasting problem is harder in this respect because it does not have the finite-state property. Here, we will investigate this problem in detail and prove that it remains hard even if one restricts to planar graphs of bounded degree or constant broadcasting time. A simple consequence is that the minimal broadcasting time cannot even be approximated with an error less than $\frac{1}{8}$, unless $\mathcal{P} = \mathcal{NP}$.

On the other hand, we will investigate for which classes of graphs this problem can be solved efficiently and show that broadcasting and even a more general version of this problem becomes easy for graphs with good decomposition properties. The solution strategy can efficiently be parallelized, too. Combining the negative and the positive results reveals the parameters that make broadcasting difficult. Depending on simple graph properties the complexity jumps from $\mathcal{NC}$ or $\mathcal{P}$ to $\mathcal{NP}$. © 1998 Elsevier Science B.V. All rights reserved.

1. Introduction

Broadcasting in processor networks means disseminating a single piece of information, which is originally known only at some nodes, called the *sources,* to all members of the network. This is done in a sequence of rounds by pairwise message exchange over the communication lines of the network. In one round each processor can send

* Corresponding author.

[1] Supported by Deutsche Forschungsgemeinschaft, Grant DFG Re672/2, and German–Israeli Foundation for Scientific Research & Development, Grant GIF I-2007-199.06/91.

e-mail: jakoby/reischuk/schindel@informatik.mu-luebeck.de.

0166-218X/98/$19.00 © 1998 Elsevier Science B.V. All rights reserved
PII S0166-218X(97)00110-8

a message to at most one of its neighbors. The goal is to inform everybody using as few rounds as possible. This number is called the *minimum broadcasting time* of the network.

Broadcasting is a basic task for multiprocessor systems that should be supported by the topology of the network. This problem has been studied extensively, mostly in the case of a single source – for an overview see [7]. In several papers the broadcast capabilities of well-known families of graphs like hypercubes, cube-connected cycles, shuffle-exchange graphs or de Bruijn graphs have been investigated and compared. In [10] Hromkovič et al. have studied the relation between the broadcasting time and the time for solving the related gossiping problem for special families of graphs.

On the other hand, one has tried to find optimal topologies for networks with a given number of nodes such that the broadcasting time is best possible. Here the worst case over all nodes as the single source should be minimized. The problem gets more complicated when restricting to graphs of bounded degree. In [13] Liestman and Peters have studied several classes of bounded degree graphs in this respect; see also [2]. Balanced binary trees already achieve a broadcasting time of logarithmic order, therefore the question is the optimal constant factor in front of the logarithm.

In this paper, we will investigate the optimization problem for arbitrary networks. That means, given a graph and a subset of nodes as sources, determine its specific broadcast time or more general find a broadcast schedule of minimal length. This problem in general is $\mathscr{NP}$-complete. We will show that this property remains even if one restricts to planar graphs of bounded degree or constant broadcast time. Furthermore, the problem cannot be solved approximately with an arbitrary precision unless $\mathscr{P} = \mathscr{NP}$.

On the other hand, we will investigate for which classes of graphs this problem can be solved efficiently. All that seems to be known is that broadcasting is easy for trees as shown by Slater et al. in [18]. Many combinatorial optimization problems for graphs have been shown to be solvable in polynomial sequential time and even in polylogarithmic parallel time for more general classes of graphs: graphs of bounded treewidth (see, for example, the paper by Arnborg et al. [1]) and graphs of small connectivity ([15] – an overview can be found in [16]).

The broadcasting problem seems to be more difficult in this respect since it does not have the finite-state property or a bounded number of equivalence classes. Thus, the methods of Arnborg et al. [1] and Reischuk [15] are not directly applicable. Still, modifying the framework developed in [15] we can show that broadcasting becomes easy for graphs with good decomposition properties. For this purpose we have to extend the notion of graph decomposition to measure its properties more exactly. A careful inspection of the possibilities how information can flow within a component and between different components of a graph will be required. For the internal flow components that are connected behave most favourably, but in general connectivity cannot always be achieved by a tree decomposition into small components. The algorithm even works for a more general version of the broadcasting problem. Furthermore, it can be parallelized efficiently to yield $\mathscr{NC}$-solutions.

As a conclusion we can say that combining these new negative and positive results the parameters that make broadcasting difficult are determined quite precisely. The complexity of this problem jumps from $\mathscr{P}$ to $\mathscr{NP}$ depending on the internal structure of the networks.

2. Definitions and previous results

A formal definition of the broadcasting problem can be given as follows.

Definition 1. Let $G=(V,E)$ be a (directed) graph with a distinguished subset of vertices $V_0 \subseteq V$, the *sources*, and $T^* \in \mathbb{N}$ be a deadline. The task is to decide whether there exists a *broadcast schedule*, that is a sequence of subsets of edges

$$E_1, E_2, \ldots, E_{T^*-1}, E_{T^*}$$

with the property $V_{T^*} = V$, where for $i>0$ we define $V_i := V_{i-1} \cup \{v \mid (u,v) \in E_i$ and $u \in V_{i-1}\}$ and require $E_i \subseteq \{(u,v) \in E \mid u \in V_{i-1}\}$ and $\forall u \in V_{i-1} : |E_i \cap (\{u\} \times V)| \leqslant 1$.

Let us distinguish between the multiple-source problem *MB* and the restricted version with only a single-source *SB*.

The meaning of the sets E_i and V_i is the following: V_i denotes the set of nodes that have received the broadcast information by round i. For $i=0$ this is just the set of sources. By the deadline T^* the set V_{T^*} should include all nodes of the network. E_i is the set of edges that are used to send information in round i, where each processor $u \in V_{i-1}$ can use at most one of its outgoing edges.

MB (denoted ND49 in [6]) has shown to be $\mathscr{NP}$-complete.

Theorem A. *MB for graphs with unbounded degree is $\mathscr{NP}$-complete, even if restricted to a fixed deadline $T^* \geqslant 4$.*

For a fixed deadline the number of sources obviously has to grow linearly in the size of the whole graph. But even the single-source problem is difficult, in this case the deadline has to grow at least logarithmically.

Theorem B. *SB for graphs with unbounded degree is $\mathscr{NP}$-complete.*

The proofs of both results were published by Slater et al. [18]. For the second result, their reduction of the three-dimensional matching problem to SB requires a deadline of order $\sqrt[3]{|V|}$ for the broadcast problem. Furthermore, in the same paper it is shown:

Theorem C. *SB can be solved in linear time for trees. This also holds for the constructive version of this problem finding an optimal broadcast schedule.*

3. New results

All theorems above can be improved significantly. For the lower bounds it suffices to consider undirected graphs, the upper bounds given below also hold for the more general case of directed graphs.

3.1. Lower bounds

Designing more complicated reductions of the three-dimensional matching problem and the 3-SAT problem we can show:

Theorem 1. *MB restricted to planar graphs with bounded degree at least* 4 *and a fixed deadline* T^* *at least* 3 *is* $\mathcal{NP}$*-complete.*

The reduction to prove this result uses graphs of a specific kind that are guaranteed to have a broadcast schedule of length 4. Now consider an approximation algorithm that for a network G gives an estimate of its minimum broadcast time $T(G)$. The estimate $\tilde{T}(G)$ may be an arbitrary real number, but is required to be within a precision γ of the correct value $\tilde{T}(G) \leqslant (1+\gamma) \cdot T(G)$. In this case, any estimate with a precision $\gamma \leqslant \frac{1}{8}$ could be used to solve the decision problem. Thus we also get:

Theorem 2. *There exists no polynomial-time approximates algorithm for MB with a precision* $\frac{1}{8}$, *unless* $\mathcal{P} = \mathcal{NP}$.

If for this minimization problem one restricts to integer values and approximations from above then the statement of the theorem can be improved from precision $\frac{1}{8}$ to any value less than $\frac{1}{3}$.

The broadcasting problem with a single source does not become substantially easier, even for bounded degree graphs with a logarithmic diameter.

Theorem 3. *SB restricted to graphs* $G = (V, E)$ *with bounded degree at least* 3 *is* $\mathcal{NP}$*-complete, even if the deadline grows at most logarithmically in the size of the graph.*

Also planarity does not make things much simpler as the following result shows.

Theorem 4. *SB restricted to planar graphs* $G = (V, E)$ *of degree* 3 *is* $\mathcal{NP}$*-complete (in this case the deadline grows like* $\sqrt{|V|}$).

For completeness, we should remark that after we have presented the lower bound in Theorem 1 the first time Middendorf was able to improve it to degree 3 and deadline 2 [14].

3.2. Upper bounds

On the positive side, we will extend the classes of graphs for which the broadcasting problem can be solved fast. For this purpose, different ways on how a graph can be decomposed into smaller components will be considered: by removing edges (edge separators) or by removing nodes (node separators). The concept of graph decomposition based on the k-connected components of a graph is developed in [8, 9]. There only node separators have been considered. For the broadcasting problem a slightly different notion of graph decomposition seems to be better suited. Furthermore, the weaker notion of edge separation is of interest because the analysis in this case is slightly less complicated and yields better bounds. For efficiency reasons an important point is to get good bounds on the round numbers, when nodes may receive the broadcast information. Things are easy if all components of the graph decomposition are connected, which in general cannot be assumed.

Here we restrict only to decompositions that generate a tree of components. Using more complicated techniques other decomposition graphs can also be handled. For the purpose of decomposing a graph G it suffices to consider only the case of undirected graphs. Thus, if G is directed in the following definition we simply mean the corresponding undirected graph.

Definition 2. A graph $H=(V_H,E_H)$ is an *edge decomposition graph* of a graph $G=(V,E)$ if the following conditions hold:
- The nodes G_i of V_H represent induced subgraphs $G_i=(V_i,E_i)$ of G such that the V_i are pairwise disjoint and $V=\bigcup_{G_i\in V_H} V_i$.
- $\{G_i,G_j\}\in E_H$ iff there is an edge between a node of G_i and a node of G_j.

H is called an *edge decomposition tree* of G if H is a tree.

Define the *cut* of an edge $\{G_i,G_j\}$, the cut of a node G_i, and the cut of H as those edges of G that connect G_i and G_j, resp. connect G_i to other components or connect any pair of components:

$$\mathrm{cut}(G_i,G_j):=\{\{u,v\}\in E \mid u\in V_i \text{ and } v\in V_j\} \quad \text{for } i\neq j,$$

$$\mathrm{cut}(G_i):=\bigcup_{\{G_i,G_j\}\in E_H} \mathrm{cut}(G_i,G_j) \qquad \text{and} \qquad \mathrm{cut}(H):=\bigcup_{G_i\in V_H} \mathrm{cut}(G_i).$$

The *border* of a node G_i are the nodes of other components that have connections to G_i:

$$\mathrm{border}(G_i):=\{v \mid \{u,v\}\in \mathrm{cut}(G_i) \text{ and } u\in V_i\}.$$

A graph $G=(V,E)$ is (κ,μ,c)-*edge decomposable* if there exists an edge decomposition graph $H=(V_H,E_H)$ such that for all $G_i\in V_H$,

$$|\mathrm{cut}(G_i)|\leqslant\kappa, \qquad |V_i|\leqslant\mu \quad \text{and} \quad \mathrm{cc}(G_i)\leqslant c,$$

where $\mathrm{cc}(G_i)$ denotes the number of connected component of G_i. In this case $\mathrm{cut}(H)$ will be called a (κ,μ,c)-*edge separator* of G.

Note that the decomposition process partitions a graph into different components. Each component G_i itself may be connected or fall into several connected components. For example, a $(\sqrt{n}\times\sqrt{n})$ – two-dimensional grid is $(\mathrm{O}(\sqrt{n}),\ \mathrm{O}(\sqrt{n}),1)$ – edge decomposable into a tree. For a cycle of length n the parameters are $(4,2,2)$. Taking the number of connected components within each component into consideration will allow us to bound the algorithmic effort to solve the broadcasting problem in a nontrivial way.

Other approaches have been proposed how to decompose a graph into smaller components, based on the notions of treewidth [17], see, for example, [1, 3, 12]. In this paper we will concentrate on the algorithmic implications of good decompositions for the broadcast problem and do not investigate the relations between the different notions of decomposibility.

In the following we assume that an edge decomposition of the network is given and do not bother how to obtain such a decomposition. Although to construct an optimal decomposition in general is $\mathcal{NP}$-complete, efficient approximation schemes are known that achieve decompositions with at most a constant factor increase in the relevant parameters.

Theorem 5. *For a graph $G=(V,E)$ of maximal degree d with a given (κ,μ,c)-edge decomposition tree MB can be solved in time*

$$\mathrm{O}\left(|V|^{c+2}\,(2(\kappa+\mu))^{\kappa+c+2}\left(2\frac{|E|+|V|}{|V|}\right)^{\mu}\right)$$

$$\leqslant \exp\mathrm{O}\left(c\log|V|+(\kappa+c)\ \log(\kappa+\mu)+\mu\log d\right).$$

The algorithm we have designed actually works for a more general version of the broadcasting problem, in which the sources may receive the broadcast information in different rounds and each node of the network may have its individual deadline. Let us call this the *general broadcasting problem GB*.

The time bound becomes polynomial for classes of graphs that can be decomposed into smaller components using not too large separators. Let *llog* and *lllog* denote the logarithm function iterated twice, (resp. three times).

Corollary 1. *Restricted to graphs $G=(V,E)$ with*

- $(\mathrm{O}(\log n/\mathrm{llog}\, n),\mathrm{O}(\log n/\mathrm{llog}\, n),\mathrm{O}(1))$-*edge decomposition trees or*
- *to graphs with bounded degree and* $(\mathrm{O}(\log n/\mathrm{llog}\, n),\mathrm{O}(\log n),\mathrm{O}(1))$-*edge decomposition trees*

MB (and even GB) can be solved in polynomial time.

So far, we have only considered the decision version of MB, resp. the task to determine the minimal length of a broadcast schedule. But applying ideas similar to the one in [15] one can also design an algorithm for constructing an optimal broadcast schedule by using the same techniques as for the decision problem.

Theorem 6. *Constructing an optimal broadcast schedule can be done in the same time bounds as stated for the decision problem in Theorem* 5.

Using the machinery developed in [15] we can also derive a fast and processor efficient parallel algorithm. Even if the decomposition tree is not nicely balanced using path compression techniques the problem can be solved with a logarithmic number of iterations (with respect to the number of components). The basic task one has to solve is how a chain of two components can be replaced by a single component that externally behaves identically with respect to broadcasting.

Theorem 7. *For a graph* $G=(V,E)$ *of maximal degree* d *with a given* (κ,μ,c)*-edge decomposition tree MB can be solved on the PRAM model in parallel time*

$$O(\log|V|\,(c\log|V|+(\kappa+c)\log(\kappa+\mu)+\mu\log d))$$

with a processor bound of $\exp O(c\log|V|+(\kappa+c)\log(\kappa+\mu)+\mu\log d)$.

For nicely decomposable classes these bounds put the MB-problem into $\mathcal{NC}$.

Corollary 2. *Restricted to graphs* $G=(V,E)$ *with*
- $(O(\log n/\text{llog}\,n), O(\log n/\text{llog}\,n), O(1))$*-edge decomposition trees or*
- *to graphs with bounded degree and* $(O(\log n/\text{llog}\,n), O(\log n), O(1))$*-edge decomposition trees*

MB is in $\mathcal{NC}^2$.

Definition 3. A graph $H=(V_H,E_H)$ is a *node decomposition graph* of a graph $G=(V,E)$ if
- the nodes G_i of V_H represent subgraphs $G_i=(V_i,E_i)$ of G such that $V=\bigcup_{G_i\in V_H} V_i$ and $E=\bigcup_{G_i\in V_H} E_i$,
- for each node v holds: if $v\in V_i\cap V_j$ then H contains a path π form V_i to V_j such that v belongs to every node V_l in π.

H is called a *node decomposition tree* of G if H is a tree. Similar as above, we define the *cut* of an edge $\{G_i,G_j\}$, of a node G_i, and of H as $\text{cut}(G_i,G_j):=V_i\cap V_j$, resp.

$$\text{cut}(G_i):=\bigcup_{\{G_i,G_j\}\in E_H}\text{cut}(G_i,G_j),\qquad \text{cut}(H):=\bigcup_{G_i\in V_H}\text{cut}(G_i).$$

The *border* of a node G_i are the nodes of other components G_j that are connected to $\text{cut}(G_i)$

$$\text{border}(G_i):=\{v\notin V_i \mid \exists u\in\text{cut}(G_i) \text{ and } \{u,v\}\in E\}.$$

A graph $G=(V,E)$ is called *(κ,μ,c)-node decomposable* if there exists a node decomposition graph $H=(V_H,E_H)$ such that for all $G_i \in V_H$ holds:

$$|\mathrm{cut}(G_i)| \leqslant \kappa, \qquad |V_i| \leqslant \mu, \quad \text{and} \quad \mathrm{cc}(G_i) \leqslant c.$$

In this case $\mathrm{cut}(H)$ is a *(κ,μ,c)-node separator* of G.

Theorem 8. *Given a graph $G=(V,E)$ of maximal degree d with a (κ,μ,c)-node decomposition tree MB can be solved in time*

$$\begin{aligned} &\mathrm{O}(|V|^{c+1}((d-1)\kappa+\mu)^{\kappa+2}(d+1)^{\mu+d\cdot\kappa}\,2^{d\cdot\kappa}) \\ &\quad \leqslant \exp \mathrm{O}(c\log|V| + \kappa\log(d\,\kappa+\mu) + (\mu+\kappa\,d)\log d). \end{aligned}$$

Similarly, we get in the parallel case:

Theorem 9. *For graphs of maximal degree d with a (κ,μ,c)-node decomposition tree MB has a parallel solution of time complexity*

$$\mathrm{O}(\log|V|\,(c\log|V| + \kappa\log(d\,\kappa+\mu) + (\mu+\kappa\,d)\log d))$$

and processor complexity $\exp \mathrm{O}(c\log|V| + \kappa\log(d\,\kappa+\mu) + (\mu+\kappa\,d)\log d)$.

As in the case of edge separators for nicely node-decomposable graphs we get:

Corollary 3. *Restricted to graphs $G=(V,E)$ with*
- *$(\mathrm{O}(1), \mathrm{O}(\log n/\mathrm{llog}\, n), \mathrm{O}(1))$-node decomposition trees, or*
- *with maximal degree $d \leqslant \mathrm{O}(\mathrm{llog}\, n/\mathrm{lllog} n)$ and $(\mathrm{O}(\log n/\mathrm{llog}\, n), \mathrm{O}(\log n/\mathrm{lllog} n), \mathrm{O}(1))$-node decomposition trees, or*
- *with constant degree and $(\mathrm{O}(\log n/\mathrm{llog}\, n), \mathrm{O}(\log n), \mathrm{O}(1))$-node decomposition trees, MB can be solved in polynomial time, even in $\mathcal{NC}^2$.*

All these bounds apply to GB as well. The constructive variant can be solved with the same effort, too.

The remaining part of this paper is organized as follows. In the next section the $\mathcal{NP}$-completeness of multiple source broadcasting in planar, bounded degree graphs is proven (Theorem 1). Section 5 describes a set of basic building blocks that are used in the lower bound proofs for single-source broadcasting. In the following two sections we give the main ideas of the reductions that yield Theorems 3 and 4. Efficient algorithms for edge-, resp. node-decomposable graphs are described in the last two sections. The reader who is interested in more details of the constructions and proofs is referred the complete technical report [11].[2]

[2] A preliminary version of these results has been presented at the 20th WG'94, Graph-Theoretic Concepts in Computer Science, Lecture Notes in Computer Science, vol. 903, Springer, Berlin, 1995, pp. 219–231.

4. MB with deadline 4 is $\mathcal{NP}$-complete

Let us first observe how a nondeterministic TM can solve MB.

Lemma 1. *MB can be solved by a NTM in time* $O(|V|^2 \log |V|)$.

Proof. For a graph $G=(V,E)$ with maximal degree d specified by adjacency lists, a set of sources V_0, and deadline $T^* \leqslant |V|$ we can solve MB by the following nondeterministic strategy:

Step 0: Generate a new graph $G' := (V, E \cup \{\{v,v\} \mid v \in V_0\})$ with maximal degree $d+1 \leqslant |V|$, that is in G' every source in G gets an additional edge.

Step 1: For each node v choose one edge $\{v',v\} \in E'$ with the interpretation that v receives the broadcast information from its neighbor v' ($v=v'$ means that v does not receive the information from somebody else).

Step 2: Let F be the subgraph of G consisting of the edges chosen in step 1. Verify that F has no directed cycle. If this condition holds F is a forest of rooted trees with edges pointing away from their roots.

Step 3: Solve the broadcasting problem for the trees constructed in step 1. Analysing the time complexity of the strategy in [18] for broadcasting in trees one can show that quadratic time suffices for this task.

The correctness follows from the fact that each broadcasting schedule can be described by a directed forest, in which the edges are labelled by the round, the broadcast information is sent across this edge. Step 1 guesses such a forest and step 3 checks whether it is possible to inform this forest within the deadline T^*. □

The $\mathcal{NP}$-hardness of MB will be proved by a reduction of 3-dimensional matching (3DM [6]):

Definition 4 (3DM). Given a set $M \subseteq A \times B \times C$, and A, B and C are disjoint sets having the same number q of elements, decide: does M contain a matching, i.e., a subset $M' \subseteq M$ such that $|M'| = q$ and no two elements of M' agree in any coordinate.

The graph $G' = G(A,B,C,M)$ of an instance (A,B,C,M) with $M \subseteq A \times B \times C$ of the 3DM problem is defined as follows: Each element of the sets A, B and C and each triple of M is represented by a vertex. The membership relation between set elements and triples defines the edges between these vertices:

$$G' := (V', E') \quad \text{with } V_A := \{\alpha_x \mid x \in A\}, \quad V_B := \{\beta_x \mid x \in B\},$$

$$V_C := \{\gamma_x \mid x \in C\}, \quad V_M := \{\mu_y \mid y \in M\}, \quad V' := V_A \cup V_B \cup V_C \cup V_M,$$

$$E' := \{(\mu_y, v_x) \mid y \in M,\ v_x \in V_A \cup V_B \cup V_C \text{ and } x \in y\}.$$

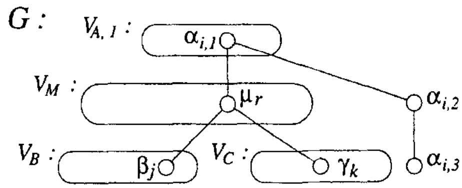

Fig. 1. The broadcasting graph corresponding to an instance of the 3DM problem.

The reduction will use a restricted version of the 3DM problem, which is still $\mathcal{NP}$-complete [5]. For an instance (A,B,C,M) of *restricted planar* $3DM$ the following properties are required:

- $G(A,B,C,M)$ is planar.
- For each element x of $A\cup B\cup C$ there are at most 3 triples in M containing x (thus, $|M|$ is bounded by $3q$ where $q:=|A|=|B|=|C|$).

Proof of Theorem 1. Let (A,B,C,M) be an instance of 3DM with $|A|=q$ and let $G'=G(A,B,C,M)$ be the matching graph. The corresponding broadcasting graph G is obtained by replacing each node $\alpha_i\in V_a$ of G' by a chain $\alpha_{i,1}$, $\alpha_{i,2}$ and $\alpha_{i,3}$ of length 3 (see Fig. 1). The other nodes and edges remain unchanged. $V_{A,1}$ is chosen as the set of sources, and the deadline is set to 3:

$$G(A,B,C,M):=(V,E) \quad \text{with } V:=V_{A,1}\cup V_{A,2}\cup V_{A,3}\cup V_B\cup V_C\cup V_M,$$

$$E:=\{(\mu_y,v_x)\mid y\in M,\ v_x\in V_A\cup V_B\cup V_C \text{ and } x\in y\}\cup\{(\alpha_{i,1},\alpha_{i,2}),(\alpha_{i,2},\alpha_{i,3})\mid i\in A\}$$

and

$$V_B:=\{\beta_x\mid x\in B\},\quad V_C:=\{\gamma_x\mid x\in C\},\quad V_{A,1}:=\{\alpha_{i,1}\mid i\in A\},$$

$$V_{A,2}:=\{\alpha_{i,2}\mid i\in A\},\quad V_{A,3}:=\{\alpha_{i,3}\mid i\in A\},\quad V_M:=\{\mu_y\mid y\in M\}.$$

Observe that $G(A,B,C,M)$ has degree 4 and is planar if G' is planar.

Lemma 2. *$G(A,B,C,M)$ has a broadcast schedule of length* 3 *iff M has a matching.*

Proof. Let $M'\subseteq M$ be a matching for A, B and C. Then the following strategy informs all nodes of $G(A,B,C,M)$ within 3 rounds:

Round 1. The q sources in $V_{A,1}$ send the information to the nodes of $V_{M'}:=\{\mu_r\mid r\in M'\}$ that represent the triples of M', hence $V_1:=V_0\cup V_{M'}$.

Round 2. The q sources inform the nodes in $V_{A,2}$. The q nodes of $V_{M'}$ informed in Round 1 inform the nodes of V_B, that menas $V_2:=V_1\cup V_{A,2}\cup V_B$.

Round 3. The nodes of $V_{A,2}$ send the information to the nodes in $V_{A,3}$, the nodes in $V_{M'}$ to the nodes of V_C, and the nodes in $V_{A,1}$ and V_B to the nodes of $V_M\setminus V_{M'}$, i.e. $V_3:=V_2\cup V_{A,3}\cup V_C\cup(V_M\setminus V_{M'})=V$.

Since M' is a matching for A, B and C the nodes in $V_{M'}$ can inform all nodes in V_B in Round 2, and all nodes in V_C in Round 3.

It is also possible to inform the nodes in $V_M \setminus V_{M'}$ in Round 3, because they can be matched with the nodes in $V_{A,1} \cup V_B$. This can be seen as follows: Each node of $V_M \setminus V_{M'}$ is connected to one node of $V_{A,1}$ and one node of V_B, whereas each node of $V_{A,1}$ and each node of V_B is connected to at most two nodes of $V_M \setminus V_{M'}$. Thus, each subset V'' of $V_M \setminus V_{M'}$ is connected to a subset of $V_{A,1} \cup V_B$ of size at least $|V''|$. Therefore, there exists a matching of $V_M \setminus V_{M'}$ with $V_{A,1} \cup V_B$.

For the other direction observe that each node $\alpha_{i,1} \in V_{A,1}$ has to inform $\alpha_{i,2}$ in the first or second round. Thus, it is only possible to inform q nodes V' of V_M by Round 1 or Round 2. These q nodes have to send the information to the $2q$ nodes of V_B and V_C. Thus, the neighborhood of V' contains all nodes of $V_{A,1}$, V_B and V_C. Since nodes in V' have degree 3, the triples corresponding to V' establish a matching M' of A, B, C. □

Theorem 2 follows easily. Since any algorithm that approximates the minimum broadcast time with a deviation smaller than $\frac{1}{8}$ has to yield a value larger then 3.5 if the optimal schedule length is 4 and smaller then 3.5 if it is 3.

5. Modular construction of difficult broadcast networks

For the single-source problem the reduction to show $\mathcal{NP}$-hardness is much more complicated. We will give a modular description by first constructing a series of some basic graphs with special broadcast properties.

Definition 5. Let a graph $G=(V,E)$ and a broadcast schedule $\mathscr{E}=E_1,E_2,\ldots$ for G be given. The first round in which a node v gets the information is called its *starting round* $t_s(v)$. If v sends the information to a neighbor in round t we call v *active* in that round. Let $t_f(v)$ be the first round by which all neighbors of v are informed. A node v is *busy* in $\mathscr{E}$ if it is active in all rounds $t_s(v)+1,\ldots,t_f(v)-1$. $\mathscr{E}$ is *busy* if all nodes are busy.

Observe that each broadcast schedule can easily be transformed into a busy broadcast schedule of the same or smaller length. Therefore, we will only consider busy broadcast schedules in the following.

The proof of Theorem 4 is based on an intricate construction of a special broadcast network G. This section precedes with an analysis of some special subgraphs that will be used as basic building blocks. Each such subgraph G' has a designated set of input and output ports. Subgraphs will be connected over these ports only. If output ports of G' are connected to input ports of another subgraph G'' we call G'' a successor of G', and G' a predecessor of G''.

If the broadcast information is sent over such a connecting edge we say that the edge is used in the corresponding round. Obviously, each edge does not have to be used more than once. The final network G will be built in such a way that in an optimal schedule the input and output edges of a subgraph G' have to be used at specific times.

Definition 6. Let, for a given schedule of G, the input edges $e_1, e_2, \ldots, e_k$ of a subgraph G' be used in rounds $t_1, t_2, \ldots, t_k$. Then the vector $\tau = \langle t_1, t_2, \ldots, t_k \rangle$ is called an *input time table* of G'. The set of all possible time tables is called the *input time sheet* $\mathscr{T}(G')$ of G'. Analogously, we define *output time tables* and *output time sheets.*

The broadcast network we are going to construct has the property that in an optimal schedule all input edges of a subgraph have to be used within a time interval of length at most 2, that means optimal input time tables are rather restricted.

Definition 7. For a subgraph G' and an input time table $\tau = \langle t_1, t_2, \ldots, t_k \rangle$ of G' let *delay*(G', τ) denote the minimal time that elapses between the round the broadcast information enters G' (that is the minimal t_i) and the first round an output edge of G' is used.

A lower bound for the time when an input edge e of G' can be used is obtained by adding up all delays on the shortest path from e to a source.

Definition 8. Let v_0 be the unique source of the broadcast network G, and let $\mathrm{path}(v_0, G')$ be the set of all paths from v_0 to the subgraph G' of G. Then define

$$\mathrm{dawn}(G') := \min_{P \in \mathrm{path}(v_0, G')} \sum_{P \text{ crosses } G'' \neq G'} \min_{\tau \in \mathscr{T}(G'')} \mathrm{delay}(G'', \tau).$$

If a node of a subgraph G' is informed in round t we call $t - \mathrm{dawn}(G')$ the *relative round* this node is informed.

Although edges between subgraphs are undirected and thus could be used in either direction we want to ensure that information enters a subgraph only at its input ports. A *ghost message* is a message that enters a subgraph G' through one of its output ports. To prevent ghost messages the following properties are helpful:
1. All successors of a subgraph G' have the same dawn.
2. All input ports of G' can be used in round $\mathrm{dawn}(G) + 2$ at the latest.
3. Let the minimal number of rounds the information needs to reach an input port of G' starting at another input port of G' and using only edges of G' the *ghost time* of G'. The ghost time of all subgraphs will be at least 3.

Let us call the mapping from the input time tables of a subgraph G' to its output time tables the *broadcast relation* $\mathscr{B}$ of G', or more formally:

Definition 9. For the set of graphs $\mathscr{G}$ described below the broadcast relation

$$\mathscr{B} : \mathscr{G} \times \mathbb{N} \times \mathbb{N}^m \to \mathsf{P}(\mathbb{N}^n) \cup \perp$$

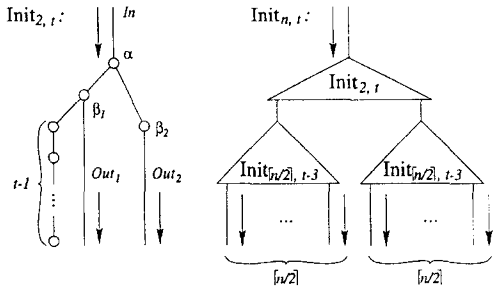

Fig. 2. The recursive construction of the initializer $\mathrm{Init}_{n,t}$.

is given by $(a_1,\ldots,a_n) \in \mathscr{B}(G',T^*,t_1\ldots t_m)$ if the following two conditions hold:

- $\tau = \langle t_1,\ldots,t_m\rangle$ is an input time table for G',
- if the m input edges of G' are used according to τ then the n output edges of G' can be used according to the output time table $\langle a_1,\ldots,a_n\rangle$ and all nodes of G' can be informed within the deadline T^*.

$\mathscr{B}(G',T^*,t_1,\ldots,t_m) = \bot$ iff it is not possible to inform all nodes of G' within the deadline T^* using the input time table τ.

Obviously, $\mathscr{B}(G',T^*,\tau)$ describes the information flow properties of G'.

5.1. Basic broadcast networks

Now we will analyze the functionality of the broadcast networks described in Figs. 2–5.

The first subgraph $\mathrm{Init}_{n,t}$ is called *initializer* (Fig. 2). If we choose the parameter $t = T^* - \mathrm{dawn}(\mathrm{Init}_{n,t})$ with $t \geqslant 3\log n - 1$ the input node α has to send the information to β_1 in round $\mathrm{dawn}(\mathrm{Init}_{n,t}) + 1$. Otherwise, the last nodes of the chain cannot receive the information within the deadline. Hence, β_1 and β_2 can inform their successors using the edges Out_1 and Out_2 simultaneously in relative round 3. The initializer transmits the information simultaneously over all its n output edges, i.e. $\mathrm{delay}(\mathrm{Init}_{n,t}, \mathrm{dawn}(\mathrm{Init}_{n,t})) = 3\log n$ and

$$\mathscr{B}(\mathrm{Init}_{n,t}, T^*, t') = \begin{cases} \{(\underbrace{3\log n,\ldots,3\log n}_{n})\} & \text{if } t' = \mathrm{dawn}(\mathrm{Init}_{n,t}), \\ \bot & \text{if } t' > \mathrm{dawn}(\mathrm{Init}_{n,t}). \end{cases}$$

The following subgraphs model a binary coding system. The two possible values correspond to a receiving the broadcast information at relative rounds 0, resp. 2.

The *guess-graph* Exist_t (Fig. 3) with $t = T^* - \mathrm{dawn}(\mathrm{Exist}_t)$ and $t \geqslant 2$ is used to generate this encoding. It holds $\mathrm{delay}(\mathrm{Exist}_t, \delta) = 2$ and

$$\mathscr{B}(\mathrm{Exist}_t, t') = \begin{cases} \{(\delta + \mathbf{2}, \delta + \mathbf{4}), (\delta + \mathbf{4}, \delta + \mathbf{2})\} & \text{if } t' = \delta, \\ \bot & \text{if } t' > \delta, \end{cases}$$

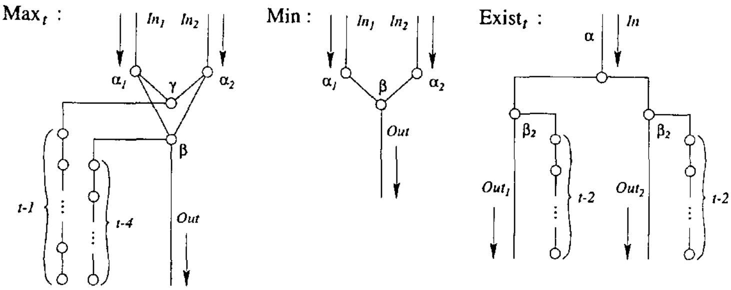

Fig. 3. The max-graph Max$_t$, the min-graph Min and the guess-graph Exist$_t$.

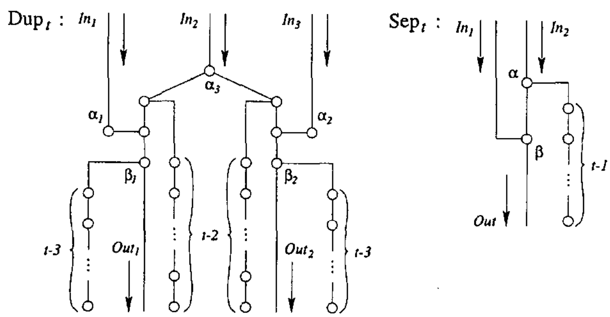

Fig. 4. The subgraphs duplicator Dup$_t$ and separator Sep$_t$.

with $\delta = \text{dawn}(\text{Exist}_t)$. Note that after informing α the broadcast strategy has to decide whether α sends the information to β_1 or β_2 first. We will interpret this decision as setting a Boolean variable.

The subgraph *duplicator* Dup$_t$ (Fig. 4) with $t = T^* - \text{dawn}(\text{Dup}_t)$ and $t \geqslant 5$ will be used to duplicates this binary encoding. The input edges In$_1$ and In$_3$ inform α_1 and α_3 in round dawn(Dup$_t$). For $\delta = \text{dawn}(\text{Dup}_t)$ it holds $\text{delay}(\text{Dup}_t, \delta, \delta, \delta) = 3$ and

$$\mathscr{B}(\text{Dup}_t, T^*, \delta, t', \delta) = \begin{cases} \{(\delta+3, \delta+3)\} & \text{if } t' = \delta, \\ \{(\delta+5, \delta+5)\} & \text{if } t' \geqslant \delta+2. \end{cases}$$

To combine two binary encodings we use the *max-graph* (Fig. 3) Max$_t$ with $t = T^* - \text{dawn}(\text{Max}_t)$ and the *min-graph* Min. It is easy to see that $\text{delay}(\text{Min}, t') = 2$. If $\min(t_1, t_2) \leqslant T^* - 2$ we get $\mathscr{B}(\text{Min}, T^*, t_1, t_2) = \{\min(t_1, t_2) + 2\}$, and else $\mathscr{B}(\text{Min}, T^*, t_1, t_2) = \perp$. The max-graph does not simulate the computation of the maximum of two input rounds precisely. If both input edges are used later than dawn(Max$_t$) at least one node $v \in \text{Max}_t$ does not receive the information within the deadline. Note that we have to guarantee that γ receives the information before $\text{dawn}(\text{Max}_t) + 1$. So we get

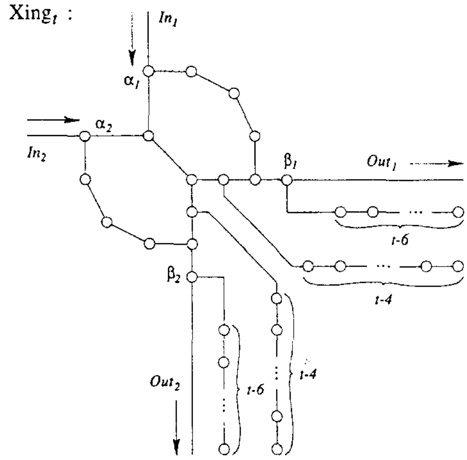

Fig. 5. The planar crossing graph Xing$_t$ with degree 3.

$\mathrm{delay}(\mathrm{Max}_t, \mathrm{dawn}(\mathrm{Max}_t), \mathrm{dawn}(\mathrm{Max}_t)) = 3$ and

$$\mathscr{B}(\mathrm{Max}_t, T^*, t_1, t_2) = \begin{cases} \{\mathrm{dawn}(\mathrm{Max}_t) + 3\} & \text{if } t_1 = t_2 = \mathrm{dawn}(\mathrm{Max}_t), \\ \{\mathrm{dawn}(\mathrm{Max}_t) + 5\} & \text{if } |\{t_i | t_i = \mathrm{dawn}(\mathrm{Max}_t),\ i \in \{1,2\}\}| = 1, \\ \perp & \text{if } t_1, t_2 > \mathrm{dawn}(\mathrm{Max}_t). \end{cases}$$

The subgraph *separator* Sep_t with $t = T^* - \mathrm{dawn}(\mathrm{Sep}_t)$ and $t \geqslant 2$ realizes a threshold function. It separates the set of all input constellations where In_1 is used at $\mathrm{dawn}(\mathrm{Sep}_t)$ into two groups:

$$\mathscr{B}(\mathrm{Sep}_t, T^*, \mathrm{dawn}(\mathrm{Sep}_t), t') = \begin{cases} \{\mathrm{dawn}(\mathrm{Sep}_t) + 1\} & \text{if } t' \leqslant \mathrm{dawn}(\mathrm{Sep}_t) + 1, \\ \{\mathrm{dawn}(\mathrm{Sep}_t) + 2\} & \text{if } t' \geqslant \mathrm{dawn}(\mathrm{Sep}_t) + 2. \end{cases}$$

Finally, the *crossing* graph Xing_t (Fig.5) with $t = T^* - \mathrm{dawn}(\mathrm{Xing}_t)$ and $t \geqslant 7$ realizes the planar crossing of one late incoming information (in round $\mathrm{dawn}(\mathrm{Xing}_t) + 2$) to the opposite output, i.e. $\mathrm{delay}(\mathrm{Xing}_t, \mathrm{dawn}(\mathrm{Xing}_t), \mathrm{dawn}(\mathrm{Xing}_t)) = 6$ and

$$\mathscr{B}(\mathrm{Xing}_t, T^*, t_1, t_2) = \begin{cases} \{(t_2 + 6, t_1 + 6)\} & \text{if } t_1 = \mathrm{dawn}(X_l) \text{ or } t_2 = \mathrm{dawn}(X_l), \\ \perp & \text{else}, \end{cases}$$

where $t_1, t_2 \in \{\mathrm{dawn}(\mathrm{Xing}_t), \mathrm{dawn}(\mathrm{Xing}_t) + 2\}$. A complete analysis of these graphs and a description of other elementary broadcasting subnetworks are given in [11].

5.2. *An exact encoding for planar crossings*

The crossing graph Xing_t does not simulate a crossing of two broadcast signals exactly. If both input edges are used late, that means at $\mathrm{dawn}(\mathrm{Xing}_t) + 2$, then there are some nodes $v \in \mathrm{Xing}_t$ that cannot received the information within the deadline. In the following we describe the basic idea of the construction of another crossing graph Cross_t that overcomes this difficulty. Two techniques are applied for this purpose.

1. The binary encoding of input rounds is made redundant by using pairs of rounds: $\mathrm{dawn}(\mathrm{Cross}_t)$ and $\mathrm{dawn}(\mathrm{Cross}_t)+2$. Such a pair can be generated by a guess-graph $\mathrm{Exist}_{t'}$. We say a schedule uses an input edge (u,v) of a subgraph G' *in time* (I) if v receives the information from u in round $\mathrm{dawn}(G)$. If v receives the information from u in round $\mathrm{dawn}(G)+2$ the schedule uses (u,v) *late* (L).
2. The crossing of two pairs $(t_1,t_2),(t_1',t_2') \in \{(I,L),(L,I)\}$ will be realized by the following strategy: We first convert both pairs into an unary coding
$$[(t_1,t_2),(t_1',t_2')] \to [(\min\{t,t'\}+c)|_{t\in\{t_1,t_2\} \text{ and } t' \in \{t_1',t_2'\}}].$$
This can be done by using several duplicators, min-graphs and crossings. In a second step we decode this unary notation to the two binary exchanged pairs by using duplicators, crossings and max-graphs. An arbitrary permutation of the positions of this unary encoding can be realized by using crossing-graphs $\mathrm{Xing}_{t''}$. The dawns of subgraphs used in the construction above are be synchronized by additional chains.

The complete crossing graph Cross_t with $t = T^* - \mathrm{dawn}(\mathrm{Cross}_t)$ and $t \geqslant 230$ can be constructed such that $\mathrm{delay}(\mathrm{Cross}_t, \delta) = 228$ and

$$\begin{aligned}&\mathscr{B}(\mathrm{Cross}_t, T^*, \delta+x, \delta+2-x, \delta+y, \delta+2-y)\\ &\quad = \{(\delta+y+228, \delta+2-y+228, \delta+x+228, \delta+2-x+228)\}\end{aligned}$$

with $x,y \in \{0,2\}$ and $d = \mathrm{dawn}(\mathrm{Cross}_t)$.

Combining some duplicators and some crossing-graphs Xing_t it is possible to construct a graph that duplicates the pairs (I,L) and (L,I). We call such a graph a *multiplicator* $\mathrm{Mult}_{n,t}$ where $t = T^* - \mathrm{dawn}(\mathrm{Mult}_{n,t})$ and n denotes the number of output pairs. This graph can be constructed such that $\mathrm{delay}(\mathrm{Mult}_{n,t}) = 39\log n$ and

$$B(\mathrm{Mult}_{n,t}, T^*, \delta+x, \delta+2-x) = \{([\delta+39\log n+x, \delta+39\log n+2-x]^n)\}$$

with $x \in \{0,2\}$ and $\delta := \mathrm{dawn}(\mathrm{Mult}_{n,t})$. All these graphs can easily be transformed into bipartite graphs with the same functionality.

6. Single-source broadcasting is $\mathcal{NP}$-complete

The $\mathcal{NP}$-hardness of the SB problem for graphs with bounded degree, will be proved by a reduction of a restricted version of 3DM problem, where for each element x of $A \cup B \cup C$ there are exactly three triples in M containing x [4]. The main idea is similar to the reduction in the proof of Theorem 1.

Consider the tree $\mathrm{Init}_{q,t}$ in Fig. 2 with its root as the only source and q outgoing edges α_i, where $t \geqslant 3\lceil \log q \rceil$. It has the following properties:

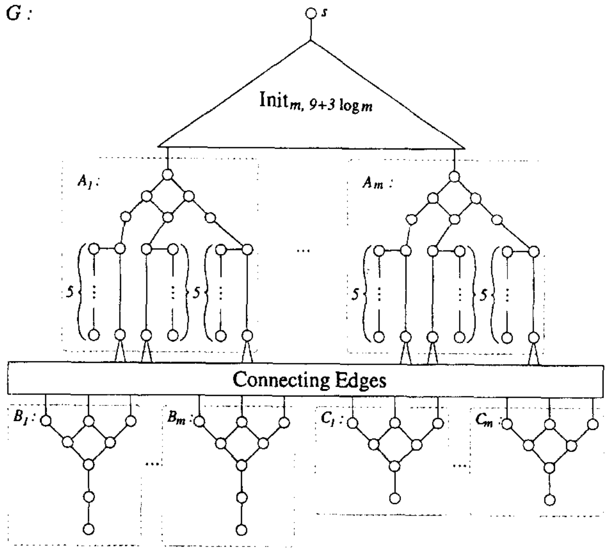

Fig. 6. A broadcast graph corresponding to an instance of the 3DM problem.

- With a delay of $\delta := 3\lceil \log q \rceil - 1$ rounds this tree can reach a state such that in the next round $\delta + 1$ the information of the source can be propagated simultaneously over all outgoing edges α_i.
- If a broadcast schedule for $\mathrm{Init}_{q,t}$ finishes by round t then none of these edges can propagate the information before round $\delta + 1$.

Connect each leaf of the tree with a node of $V_{A,1}$ of the graph G defined above. Let $t := 3\lceil \log q \rceil + 3$ and connect the root of $\mathrm{Init}_{q,t}$ with the source v_0. Then this new graph G' has a broadcast schedule of length at most $3\log q + 4$ iff M contains a matching. The resulting graph has degree 5, but is not necessarily planar since edges from $V_{A,1}$ to V_M may have to be crossed by the edges leaving $\mathrm{Init}_{q,t}$. By additional effort G' can be modified to decrease the node degree to 3.

Observe that a graph $G = (V, E)$ with a single source cannot be informed within less then $\log |V|$ rounds. Our construction yields that the problem to find a minimal broadcast is $\mathcal{NP}$-complete for logarithmic deadlines.

Proof of Theorem 3. Let A, B, C, M be an instance of 3DM with $|A| = q$. The corresponding graph G with unique source s consist of 4 levels (Fig. 6):

The first level consists of the source s connected with the root of an initializer that duplicates the number of sources. The second level consists of some subgraphs A_i simulate the nodes of $V_{A,1}$ and V_M in the proof of Theorem 1. The third level consists of the edges connecting a leaf of the subgraph A_i with an input node of the subgraph B_j and an input node of C_k iff $(i,j,k) \in M$. This means that the leaves of the subgraphs A_i simulate the nodes of V_M. The fourth level consists of some subgraphs B_j and C_k

that simulate the nodes of V_B and V_C. The deadline is chosen as $T^* := 10 + 3\log q$. Observe that G has maximal degree 3.

Lemma 3. *Let $(G=(V,E), V_0, T^*)$ represent an instance $M \subseteq A \times B \times C$ of the restricted 3DM-problem. Then G can be informed within T^* rounds iff M contains a matching.*

Proof. Let $V' := V \setminus (\bigcup_{i=1}^{p}(V_{B_i} \cup V_{C_i}))$ and $G' := (V', E')$ with $E' := \{(u,v) \in E \mid u, v \in V'\}$. Then for each schedule S' for G' with deadline T^* holds: At most one leaf of each A_i receives the information in round $T^* - 5$ and at least two leaves in a round $T^* - 3$ or later. So the claim follows similar to the proof of Lemma 2. □

7. SB of planar graphs is $\mathcal{NP}$-complete

To achieve planarity in the single source case we construct a direct reduction of 3SAT. Although there are planar versions of 3SAT that remain $\mathcal{NP}$-complete they do not help much in this case because the connections to the source will destroy planarity. A simple exchange of an edge crossing by a planar subgraph with 2 inputs and 2 outputs as given by the subgraph Xing in Fig. 5 does not seem to work.

We have found a way to allow such a replacement under special circumstances, namely if the direction of the information flow over the edges is known in advance and if at most one of the two input edges is used in time. The first property can easily be achieved for 3SAT, while the second requires special codings.

The reduction will use the following restricted version of the satisfiability problem: Let F be an Boolean formula such that for each variable $x_i \in U$ there are at most 5 clauses in F that contain either x_i or $\overline{x_i}$ and each clause $C_i \in F$ satisfies $|C_i| = 3$. This restricted version of 3SAT remains $\mathcal{NP}$-complete [6].

Proof of Theorem 4. We reduce a given instance F of the restricted version of 3SAT with clauses $C_1 \dots C_m$ and variables $x_1 \dots x_n$ to a graph consisting of 5 levels (Fig. 7).

1. The first level consists of the source s connected with the root of an initializer $\mathrm{Init}_{n,t}$ with $t := T^* - 1$.
2. The leaves of the initializer are connected with the inputs of n parallel guess-graphs $\mathrm{Exist}_{t'}$ with $t' := T^* - 3\log n$. Let G_1 be the subgraph consisting of the source, the initializer and the guess-graphs. Let S be an arbitrary schedule for G_1 achieving the deadline such that the output edges Out_1 and Out_2 of the guess-graphs (Fig. 3) can be used without artificial delay. Then one of the edges $\mathrm{Out}_1, \mathrm{Out}_2$ can be used in round $3\log n + 3$ and the other one in round $3\log n + 5$. We will denote this behaviour by the time tuples $(0,2)$ or $(2,0)$ relative to the dawn of the successors, and interpret these pairs as codings for true and false setting of the corresponding variable.
3. The third level consists of n parallel multiplicators $\mathrm{Mult}_{5,t''}$, which are used to increase the number of binary encodings chosen in level 2.

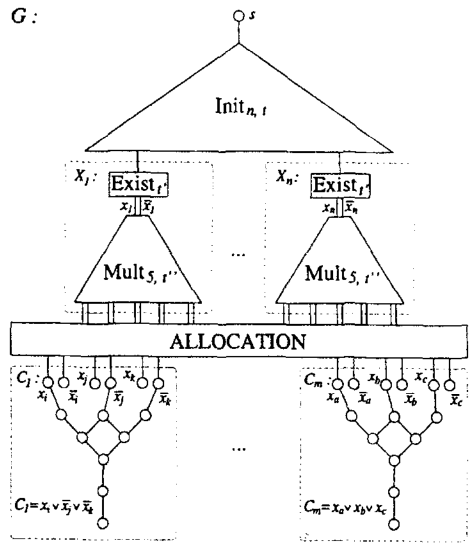

Fig. 7. A planar broadcasting graph corresponding to an instance of the restricted version of 3SAT.

4. On this level we have *send* these binary encodings to the subgraphs of the last level which represent the clauses $C_1, \ldots, C_m$. In this network we will use a special coding and decoding network to realize a crossing of the relative time tuples $(0,2)$ and $(2,0)$. We call a pair of coding and decoding network a double crossing. These components are combined in an allocation network depth $228(5n)$ and size $O(n^2)$.
5. Finally, we connect the output nodes of the allocation network to the OR-graphs C_i.

For the resulting graph G with source s the deadline is set to $\text{dawn}(C_1) + 4$, i.e.

$$T^* := 3\log n - 1 + 39 \cdot 3 + 2 + 228 \cdot d + 5.$$

Lemma 4. *Let (G, V_0, T^*) represent an instance F of the restricted version of 3SAT. Then G can be finished within T^* rounds iff there is a satisfying truth assignment for F.*

Proof. The claim follows from the fact that a schedule can only achieve the deadline iff for each subgraph C_i there exists at least one input node v that is connected to an inner node of C_i and v receives the information in round $T^* - 5$. □

8. Efficient algorithms for decomposable graphs

We start with a generalization of the broadcast problem. So far, each source node has got the broadcast information in round 0. In the more general case, a source v may get the information in an arbitrary round $\sigma(v) \geqslant 0$. Furthermore, for each node v there is an individual deadline $\rho(v)$, instead of a global deadline T^* identical for all nodes. This generalization may be of less interest with respect to practical applications. Nevertheless, it is necessary in order to apply an approach based on graph decompositions, as it has been for several other graph theoretical decision and optimization problems.

Definition 10 (*GENERAL BROADCAST-problem* [*GB*]). Given a graph $G=(V,E)$ and two partial functions $\sigma,\rho : V \longrightarrow \mathbb{N}$, decide whether there exists a broadcast schedule $E_1, E_2, \ldots$ with

$$V_i = V_{i-1} \cup \{ v \mid (u,v) \in E_i \wedge u \in V_{i-1} \} \cup \{ v \in V \mid \sigma(v) = i \},$$
$$E_i \subseteq \{(u,v) \in E \mid u \in V_{i-1}\} \quad \text{and} \quad \forall u \in V_{i-1} : |E_i \cap (\{u\} \times V)| \leqslant 1$$

such that $\forall v \in V : v \in V_{\rho(v)}$ if $\rho(v)$ is defined.

The set of sources V_s are given by the domain of σ.

The GB-problem can be solved similarly to the strategy of Lemma 1. Thus, this problem is also $\mathcal{NP}$-complete. If we restrict the GB-problem to graphs $G=(V,E)$ with maximal degree d the number of different choices of step 1 is bounded by

$$\prod_{v \in V} d_v \leqslant \left(\frac{2|E'|}{|V|} \right)^{|V|} \leqslant (d+1)^{|V|},$$

where d_v denotes the degree of $v \in V$ in G'. Thus for a graph $G=(V,E)$ the GB-problem can be solved in time $\mathrm{O}(|V|^2(2(|E|+|V|)/|V|)^{|V|})$. Let V' denote the set of nodes of degree 1 in G, then restricted to a node $v \in V'$ we can simplify step 1 of the algorithm given in the proof of Lemma 1 as follows: if v is a source with $\sigma(v) \leqslant \rho(v)$ we choose the edge $\{v', v\}$, and $\{v, v'\} \in E$ else. Thus, there are at most

$$\prod_{v \in V \setminus V'} \delta'(v) \leqslant \left(2 \frac{|E| + |V| - |V'|}{|V| - |V'|} \right)^{|V|-|V'|} \leqslant (d+1)^{|V|-|V'|}$$

different choices.

Lemma 5. *Let V' denote the set of nodes of G of degree 1. Then the GB-problem for G can be solved in time*

$$\mathrm{O}\left(|V|^2 \left(2 \frac{|E|+|V|-|V'|}{|V|-|V'|} \right)^{|V|-|V'|} \right).$$

The above strategy can be parallelised in a simple way.

Lemma 6. *The GB-problem restricted to graphs $G=(V,E)$ with maximum degree d can be solved by a* CRCW-PRAM *with $\mathrm{O}((2(|E|+|V|)/|V|)^{|V|})$ processors in time $\mathrm{O}(|V|^2)$.*

These strategies will be used as basic routines for the components of a graph.

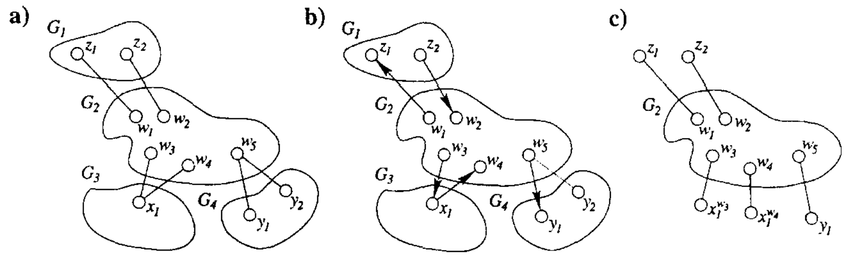

Fig. 8. (a) A node G_2 of an edge decomposition tree and its neighbors. (b) A possible information flow within a broadcast schedule: from G_2 to other components in one or in both directions, some edges may not be used. (c) The minimum deadline of the general broadcast problem for this graph is used to calculate the minimum broadcast time for the graph above.

Proof of Theorem 5. Let $H=(V_H,E_H)$ be a (κ,μ,c)-edge decomposition tree of a graph $G=(V,E)$ with $V_H=\{G_1,\dots,G_k\}$. Fig. 8(a) shows such a component G_2, which is connected to three components G_1, G_3 and G_4. A component generated by H may fall into several connected subgraphs which we will call *subcomponents*.

Let $\mathscr{G}_i:=\{G_i^1,\dots,G_i^{c_i}\}$ with $\boldsymbol{G_i^a}=(V_i^a,E_i^a)$ be the set of subcomponents of the component G_i and define $\operatorname{cut}(G_i^a)$ as the set of edges of $\operatorname{cut}(G_i)$ with one endpoint in G_i^a. Define

$$\operatorname{cut}(G_i^a,G_j):=\operatorname{cut}(G_i,G_j)\cap\operatorname{cut}(G_i^a),$$

$$\operatorname{cut}(G_i^a,G_j^b):=\operatorname{cut}(G_i^a,G_j)\cap\operatorname{cut}(G_j^b),$$

$$\operatorname{border}(G_i^a):=\{v\mid \{u,v\}\in\operatorname{cut}(G_i^a)\text{ and }u\in V_i^a\}.$$

To describe a broadcasting schedule $\mathscr{E}$ of G, each edge $\{u,v\}$ of G is labelled by $(\tau(u,v),r(u,v))$. The first value $\tau(u,v)$ denotes the round this edge is used and the second $r(u,v)$ the direction ($[u\to v]$ or $[v\to u]$). If this edge is not used we set $\tau(u,v):=-1$.

If we restrict $\mathscr{E}$ to $\operatorname{cut}(G_i^a)$ it suffices to denote the first round $\boldsymbol{\tau(G_i^a)}$ a node of G_i^a gets the broadcast information and for each edge $\{u,v\}$ with $\tau(u,v)\geq 0$ the *relative round*

$$\boldsymbol{\widehat{\tau}(u,v)}:=\tau(u,v)-\tau(G_i^a).$$

If the edge $\{u,v\}$ is not used we set $\widehat{\tau}(u,v):=-1$. Let $\boldsymbol{\tau(G_i^a,G_j^b)}$ be the first round an edge of $\operatorname{cut}(G_i^a,G_j^b)$ is used. If no edge in $\operatorname{cut}(G_i^a,G_j^b)$ is used $\tau(G_i^a,G_j^b):=-1$. Similarly, define

$$\boldsymbol{\widehat{\tau}(G_i^a,G_j^b)}:=\tau(G_i^a,G_j^b)-\tau(G_i^a)\qquad\text{if }\tau(G_i^a,G_j^b)\geq 0,\text{ else let }\widehat{\tau}(G_i^a,G_j^b):=-1.$$

The following two lemmas show that with the help of the concept of relative rounds $\widehat{\tau}$ the number of possible protocols of information exchange between two components can be bounded quite substantially. This property will be basic for the time efficiency of the algorithm.

Lemma 7. *If $\mathscr{E}$ is a busy broadcast schedule then for all $\{u,v\}\in \mathrm{cut}(G_i^a, G_j^b)$ holds: the numbers $\widehat{\tau}(u,v)$ and $\widehat{\tau}(G_i^a, G_j^b)$ are smaller than $|\mathrm{cut}(G_i^a)| + |V_i^a| - 1 \leqslant \kappa + \mu - 1$.*

Proof. Let $G_i' = (V_i^a \cup \mathrm{border}(G_i^a), E_i^a \cup \mathrm{cut}(G_i^a))$ be the extended component of $G_i^a = (V_i^a, E_i^a)$. Note that the number of nodes of G_i' is bounded by $\kappa + \mu$. Thus, the minimal broadcasting time of G_i' is trivially bounded by $\kappa + \mu - 1$.

The claim follows from the fact, that $\widehat{\tau}(G_i^a, G_j^b) + \widehat{\tau}(u,v)$ denotes the delay between the first node of G_i^a being informed and the round when the information is sent across $\{u,v\}$. □

For neighboring components G_i, G_j define a *relative state* as a tuple

$$\gamma_{i,j} := [(\widehat{\tau}(e), r(e)) \mid e \in \mathrm{cut}(G_i, G_j)]$$

Fig. 8(b) illustrates a complex information flow between a component and its neighbors.

The *relative surface* $\Gamma_{i,j}^{\mathrm{r}}$ is the set of all possible relative states $\gamma_{i,j}$ of busy broadcast schedules. A *state* $S_{i,j}$ between two neighbors G_i and G_j is a vector consisting of a relative state $\gamma_{i,j}$ and a starting round $\tau(G_i^a)$ for all subcomponents of G_i with $\mathrm{cut}(G_i^a, G_j) \neq \emptyset$. Let $\boldsymbol{\Gamma_{i,j}}$ be the set of all possible states $S_{i,j}$ that may appear in busy schedules. A *state* S_i of a component G_i is a vector consisting of the starting round $\tau(G_i^a)$ for all subcomponents of G_i and tuples $\gamma_{i,j}$ for all neighboring components of G_i. As above, let $\boldsymbol{\Gamma_i}$ be the set of all possible states S_i that may appear in busy schedules.

Lemma 8. *For a component G_i with cutsize $|\mathrm{cut}(G_i)| \leqslant \kappa_i$, size $|V_i| \leqslant \mu_i$ and $c_i = \mathrm{cc}(G_i)$ subcomponents, the size of Γ_i is bounded by*

$$\gamma(\kappa_i, \mu_i, c_i) := |V|^{c_i}(2(\kappa_i + \mu_i))^{\kappa_i}.$$

Proof. $|\Gamma_i| \leqslant \prod_{G_i^a \in \mathscr{G}_i} |V| \; \prod_{\mathrm{cut}(G_i,G_j)\neq\emptyset} |\Gamma_{i,j}| \leqslant |V|^{c_i}(2(\kappa_i + \mu_i))^{\kappa_i}$. □

Note that $|\Gamma_{i,j}| \leqslant |\Gamma_i|$.

The following strategy solves the minimum broadcasting time problem for graphs $G = (V, E)$ with given (κ, μ, c)-edge decomposition tree $H = (V_H, E_H)$. Let $\boldsymbol{\Delta(G_i, S_i)}$ denote the minimal schedule length of the local broadcast problem for the graph G_i and external information exchange as specified by state S_i ($=\infty$ if there is no schedule for state S_i). Observe that this value is independent of the structure of G outside of G_i.

Step 1: For each component $G_i = (V_i, E_i)$ and each state $S_i \in \Gamma_i$ determine $\Delta(G_i, S_i)$. For each edge $\{v, w\} \in \mathrm{cut}(G_i)$ with $w \notin G_i$ and $\widehat{\tau}(v, w) \geqslant 0$ generate a new node $w^{[v]}$ and a new edge $\{w^{[v]}, v\}$, where v is the only neighbor of $w^{[v]}$. Define $V(S_i)$ as the set of these new nodes $w^{[v]}$ and $E(S_i)$ the set of new edges $\{w^{[v]}, v\}$. We define a local GB-problem with respect to G_i and S_i as follows:

$$G_i' := (V_i \cup V(S_i), E_i \cup E(S_i)),$$
$$\sigma^i(w^{[v]}) := \tau(v,w) - 1 \quad \text{if } S_i r(v,w) = [w \to v],$$
$$\rho^i(w^{[v]}) := \tau(v,w) \quad \text{if } S_i r(v,w) = [v \to w].$$

For $v \in V_i$ define $\sigma^i(v) := \sigma(v)$ and $\rho^i(v) := \rho(v)$.

The construction of G_i' for the example given in Fig. 8(b) is illustrated in Fig. 8(c).

Step 2: Let $G_{i,0}, \ldots, G_{i,\ell_i}$ denote the neighbors of G_i. Choose an arbitrary component G_r and declare G_r as the root of H. Let $G_{i,0}$ be the father of G_i in H according to the orientation with respect to G_r. Let G_i^* denote the subgraph of G containing G_i and all its descendents. Evaluate the function $\Delta(G_i^*, S_{i,0})$ for all G_i and $S_{i,0}$ starting with the leaf components of H.

Lemma 9. *Let $G_{i,1}, \ldots, G_{i,\ell_i}$ denote the sons of G_i and let $S_{i,j}$ be a state connecting G_i and $G_{i,j}$. The minimal deadline for the general broadcast problem for G_i^* with respect to external information exchange $S_{i,0}$ can be computed as*

$$\Delta(G_i^*, S_{i,0}) = \min_{\substack{S_i = (\tau_1, \ldots, \tau_{\mathrm{cc}(G_i)}, \gamma_{i,0}, \ldots, \gamma_{i,\ell_i}) \in \Gamma_i \\ \text{with } S_{i,0} \subseteq S_i}} \max\left(\left\{ \min_{S_{j,i} \in \Gamma_{j,i}(S_i)} \Delta(G_{i,j}^*, S_{j,i}) \mid j \in [1 \ldots \ell_i] \right\} \cup \{\Delta(G_i, S_i)\} \right)$$

with

$$\begin{aligned} \Gamma_{j,i}(S_i) := \{ ([\tau(G_{j,i}^b) = \tau_k + \hat{\tau}(G_i^k, G_{j,i}^b) - \hat{\tau}(G_{j,i}^b, G_i^k) \mid \mathrm{cut}(G_i^k, G_{j,i}^b) \neq \emptyset], \\ [(\hat{\tau}(e) - \hat{\tau}(G_i^k, G_{j,i}^b) + \hat{\tau}(G_{j,i}^b, G_i^k), r(e)) \mid e \in \mathrm{cut}(G_i^k, G_{j,i}^b) \textit{ and} \\ (\hat{\tau}(e), r(e)) \in \gamma_{i,j}]) \}. \end{aligned}$$

Proof. This property can be shown by induction on the depth of the subgraphs G_i.

If G_i is a leave of H then $S_i = S_{i,0}$ thus $\Delta(G_i^*, S_{i,0}) = \Delta(G_i, S_i)$. The claim follows directly from the definition of $\Delta(G_i, S_i)$.

Let h_i be the depth of the subgraphs G_i. Assume that the claim holds for each subtree of depth less than h_i and each state of these subtrees, in particular for each son $G_{i,k}$ of G_i and each state $S_{k,i}$.

Given a state $S_i = (\tau_1, \ldots, \tau_{\mathrm{cc}(G_i)}, \gamma_{i,0}, \ldots, \gamma_{i,\ell_i}) \in \Gamma_i$ of G_i then $\Gamma_{j,i}(S_i)$ denotes the set of corresponding state $S_{j,i}$. Thus, $\Delta(G_j^*, S_{j,i})$ denotes the minimal deadline of G_j^* with respect to external information exchange $S_{j,i}$ and

$$\max\left(\max_{j \in [1 \ldots \ell_i]} \left\{ \min_{S_{j,i} \in \Gamma_{j,i}(S_i)} \Delta(G_{i,j}^*, S_{j,i}) \right\} \cup \{\Delta(G_i, S_i)\} \right)$$

the minimal deadline of G_i^* with respect to external information exchange S_i.

The claim of the lemma follows since we minimize over all possible states of G_i that may appear in busy schedules. $\square$

Therefore, $\Delta(G_r^*) = \Delta(G_r^*, \lambda)$ denotes the minimal schedule length for the graph G itself. The correctness of step 1 follows directly from the definition of a surface and the definition of the general broadcast problem. The correctness of step 2 follows from Lemma 9.

According to Lemma 5 the computation of $\Delta(G_i, S_i)$ requires at most $O((\mu_i + \kappa_i)^2(2(|E| + |V|)/|V|)^{\mu_i})$ steps. From Lemma 8 follows that step 1 can be executed in time

$$\sum_{G_i} \gamma(\kappa_i, \mu_i, c_i) O\left((\mu + \kappa)^2 \left(2\frac{|E|+|V|}{|V|}\right)^{\mu}\right)$$
$$\leqslant O\left(|V|^c (2(\kappa + \mu))^{\kappa} (\mu + \kappa)^2 \left(2\frac{|E|+|V|}{|V|}\right)^{\mu}\right).$$

The computation of $\Delta(G_i^*, S_{i,0})$ is independent of the remaining structure of G. Note that given S_i $\widehat{\tau}(G_i^a, G_j^b) = \min_{e \in \mathrm{cut}(G_i^a, G_j^b)} \widehat{\tau}(e)$, so $|\Gamma_{i,j}(S_i)| \leqslant (\mu + \kappa)^c$. Thus given all values $\Delta(G_i, S_i)$ and $\Delta(G_{i,j}^*, S_{j,i})$ the computation of all $\Delta(G_i^*, S_{i,0})$ can be executed in time

$$O(\gamma(\kappa_i, \mu_i, c_i) \ell_i (\mu + \kappa)^c c^2).$$

Summing up over all G_i gives the bound

$$\sum_{G_i} O(\gamma(\kappa_i, \mu_i, c_i) \ell_i (\mu + \kappa)^c c^2) \leqslant O\left(\gamma(\kappa, \mu, c)(\mu + \kappa)^c c^2 \sum_{G_i} \ell_i\right)$$
$$\leqslant O(|E| \gamma(\kappa, \mu, c)(\mu + \kappa)^c c^2).$$

This finishes the proof of Theorems 5 and 6. □

By using tree contraction methods the evaluation of the Δ-function can also be done in parallel requiring only a logarithmic number of iterations which yields Theorem 7. The details are described in [15].

9. Node separation

The same technique with a slightly worse-time bound due to a larger number of states also works for node decompositions of graphs.

Proof of Theorem 8. Let $H = (V_H, E_H)$ be a (κ, μ, c)-node decomposition tree of the graph $G = (V, E)$ with $V_H = \{G_1, \ldots, G_k\}$. Fig. 9(a) shows a component G_2 which is connected to three other components G_1, G_3, and G_4.

Let $\mathscr{G}_i := \{G_i^1, \ldots, G_i^{c_i}\}$ with $G_i^a = (V_i^a, E_i^a)$ be the set of subcomponents of the G_i and define $\mathrm{cut}(G_i^a)$ as the set of nodes of $V_i^a \cap \mathrm{cut}(G_i)$:

$$\mathrm{cut}(G_i^a, G_j) := \mathrm{cut}(G_i, G_j) \cap \mathrm{cut}(G_i^a),$$
$$\mathrm{cut}(G_i^a, G_j^b) := \mathrm{cut}(G_i^a, G_j) \cap \mathrm{cut}(G_j^b),$$
$$\mathrm{border}(G_i^a) := \{v \notin V_i^a \mid \{v, u\} \in E \text{ and } u \in \mathrm{cut}(G_i^a)\}.$$

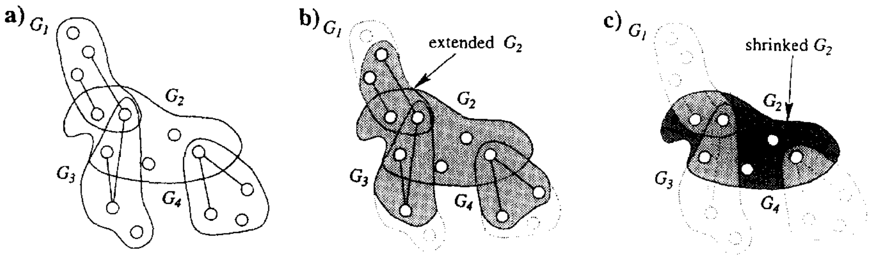

Fig. 9. (a) A node G_2 of a edge decomposition tree and its neighbors. (b) The extended component of G_2. (c) The minimum deadline of the general broadcast problem of this graph is used to calculate the minimum broadcast time for the graph above.

For a node $u \in V$ let $\tau(u)$ be the round u gets the information, and for an edge $e_l \in E$ let $\tau(e_l)$ the round when e_l is used. To describe a broadcasting schedule of a graph G, each node u of G is labelled by a vector of rounds $\gamma_u := (\tau(u), \hat{\tau}(e_1), \dots, \hat{\tau}(e_{\eta(u)}))$, where $e_1, \dots, e_{\eta(u)}$ denote the edges which are incident to u and $\boldsymbol{\hat{\tau}(e_l)} := \tau(e_l) - \tau(u)$. Note that the values $\hat{\tau}(e_l)$ are bounded by d. γ_u will be called the *state* of u. The *surface* Γ_u of a node $u \in V$ is the set of all possible states of u that may appear in a busy broadcast schedules. Note that for a fixed $\tau(u)$ $|\Gamma_u|$ is bounded by $(d+1)^d$.

For a node $u \in V_i^a$ let $\boldsymbol{\hat{\tau}(u)} := \tau(u) - \tau(G_i^a)$ where $\tau(G_i^a)$ denotes the first round a node of V_i^a receives the information.

Lemma 10. *For a busy broadcast schedule $\hat{\tau}(u)$ is bounded by $(d-1)\kappa + \mu - 1$.*

Proof. For a component $G_i^a = (V_i^a, E_i^a)$ with $|V_i^a| \geqslant 2$ let $G_i' = (V_i^a \cup \text{border}(G_i^a), E_i^a \cup \{\{u,v\} \in E \mid u \in \text{cut}(G_i^a)\})$ be the extended component of G_i^a (see Fig. 9(b)). Note that the number of nodes of $G_i' = (V_i', E_i')$ is bounded by $(d-1)\kappa + \mu$. Thus, the minimum broadcast time of G_i' is bounded by $(d-1)\kappa + \mu - 1$.

Let u and v be two nodes of a subcomponent of a component of G. Then the difference between the rounds when the two nodes u and v are informed is at most $(d-1)\kappa + \mu - 1$. □

A *state* $\gamma_{i,j}$ between two neighbors G_i and G_j is a vector of states γ_u, one for each node of $\text{cut}(G_i, G_j)$. The *surface* $\Gamma_{i,j}$ of a $\text{cut}(G_i, G_j)$ is the set of all possible states $\gamma_{i,j}$ that may appear in busy broadcast schedules. A *state* S_i of a component G_i is a vector of states γ_u, one for each node of $\text{cut}(G_i)$. The *surface* Γ_i of a component G_i is the set of all possible states S_i that may appear in busy schedules.

Lemma 11. *For a component G_i with cutsize $|\text{cut}(G_i)| \leqslant \kappa_i$, size $|V_i| \leqslant \mu_i$ and $c_i = \text{cc}(G_i)$ subcomponents, the size of Γ_i is bounded by $\gamma(\kappa_i, \mu_i, c_i) := |V|^{c_i}(((d-1)\kappa_i + \mu_i)(d+1)^d)^{\kappa_i}$.*

Proof. Define the *relative state* $\gamma_u^{\mathrm{r}} := (\hat{\tau}(u), \hat{\tau}(e_1), \dots, \hat{\tau}(e_{\eta(u)}))$ and the *relative surface* Γ_u^{r} as the set of all possible relative states γ_u^{r} that may appear in busy broadcast schedules. Then the *state* $\overline{\gamma}_i$ between two neighbors G_i and G_j is a vector

$$\overline{\gamma}_i := [\tau(G_i^a) \mid G_i^a \in \mathcal{G}_i], [\gamma_u^{\mathrm{r}} \mid u \in \mathrm{cut}(G_i)].$$

The *surface* $\overline{\Gamma_i}$ of $\mathrm{cut}(G_i)$ is the set of all possible states $\overline{\gamma}_i$ that may appear in busy broadcast schedules. Note that $|\overline{\Gamma_i}| = |\Gamma_i|$ and $|\Gamma_{i,j}| \leqslant |\Gamma_i|$. Hence,

$$|\Gamma_i| \leqslant \prod_{G_i^a \in \mathcal{G}_i} |V| \prod_{u \in \mathrm{cut}(G_i^a)} |\Gamma_u^{\mathrm{r}}| \leqslant |V|^{c_i}(((d-1)\kappa_i + \mu_i)(d+1)^d)^{\kappa_i}. \qquad \square$$

The following strategy solves the minimum broadcasting time problem for graphs $G=(V,E)$ with given (κ,μ,c)-node decomposition tree $H=(V_H,E_H)$. Let $\boldsymbol{\Delta(G_i, S_i)}$ denote the minimal schedule length of the local broadcast problem for the graph G_i and external information exchange as specified by state S_i ($=\infty$ if there is no schedule for state S_i). Again this value is independent of the structure of G outside of G_i.

Step 1: For each component $G_i=(V_i,E_i)$ and each state $S_i \in \Gamma_i$ determine $\Delta(G_i,S_i)$. Let $G_i' := (V_i \backslash \mathrm{cut}(G_i), E_i \backslash \{\{u,v\} | u \in \mathrm{cut}(G_i)\})$ be the shrunken component of G_i. For each edge $\{v,w\} \in E_i$ with $v \in \mathrm{cut}(G_i)$ generate a new node $w^{[v]}$ and a new edge $\{w^{[v]},v\}$, such that v is the only neighbor of $w^{[v]}$. Define $V(S_i)$ as the set of these new nodes $w^{[v]}$, and $E(S_i)$ the set of new edges $\{w^{[v]},v\}$. We define a local GB-problem with respect to G_i and S_i as follows:

$$G_i'' := (V_i' \cup V(S_i), E_i' \cup E(S_i)),$$
$$\sigma^i(w^{[v]}) := \tau(v,w) - 1 \quad \text{if } \gamma_w \in S_i \hat{\tau}(v,w) > 0,$$
$$\rho^i(w^{[v]}) := \tau(w) \quad \text{if } \gamma_u \in S_i \hat{\tau}(v,w) = 0.$$

For $v \in V_i'$ define $\sigma^i(v) := \sigma(v)$ and $\rho^i(v) := \rho(v)$.
Note that $|V_i''| \leqslant \mu_i + (d-2)\kappa_i$. The construction of G_i'' is illustrated by Fig. 9(c).

Step 2: Let $G_{i,0},\ldots,G_{i,\ell_i}$ denote the neighbors of G_i. Choose an arbitrary component G_r and declare G_r as the root of H. Let $G_{i,0}$ be the father of G_i in H according to the orientation with respect to G_r. Let G_i^* denote the subgraph of G containing G_i and all its descendants. Evaluate the function $\Delta(G_i^*,S_{i,0})$ for all G_i and $S_{i,0}$ starting with the leaf components of H.

Lemma 12. *Let $G_{i,1},\ldots,G_{i,\ell_i}$ denote the sons of G_i and let $S_{i,j}$ be a state connecting G_i and $G_{i,j}$. The minimal deadline for the general broadcast problem for G_i^* with respect to external information exchange $S_{i,0}$ can be computed as*

$$\Delta(G_i^*,S_{i,0}) = \min_{\substack{S_i=(\gamma_u | u \in \mathrm{cut}(G_i)) \in \Gamma_i \\ \text{with } S_{i,0} \subseteq S_i}} \max(\{\Delta(G_{i,j}^*,S_{j,i}) \mid j \in [1\ldots l_i]\} \cup \{\Delta(G_i,S_i)\})$$

with $S_{j,i} := (\gamma_u \mid u \in \mathrm{cut}(G_i,G_{i,j})) \subseteq S_i$.

The proof is almost identical to the one of Lemma 9.

As in the case of edge-decomposition $\Delta(G_r^*) = \Delta(G_r^*,\lambda)$ denotes the minimal schedule length for the graph G itself.

The correctness of step 1 follows directly from the definition of a surface and the correctness of step 2 from Lemma 12. According to Lemma 5 the computation of $\Delta(G_i, S_i)$ requires at most $O(((d-2)\kappa_i + \mu_i)^2(d+1)^{\mu_i})$ steps. From Lemma 11 follows that step 1 can be executed in time

$$\sum_{G_i} \gamma(\kappa_i, \mu_i, c_i) O(((d-1)\kappa + \mu)^2 (d+1)^{\mu}) \leqslant O(|V|^c((d-1)\kappa + \mu)^{\kappa+2}(d+1)^{\mu+d\kappa}).$$

The computation of $\Delta(G_i^*, S_{i,0})$ is independent of the remaining structure of G. Note that for fixed $S_{i,0}$ the number for $S_{i,j}$ that may appear in a busy broadcast schedule is bounded by $2^{(d-1)\kappa}$. Thus, given all values $\Delta(G_i, S_i)$ and $\Delta(G_{i,j}^*, S_{j,i})$ the computation of all $\Delta(G_i^*, S_{i,0})$ can be executed in time $O(\gamma(\kappa_i, \mu_i, c_i)\ell_i 2^{(d-1)\kappa})$. Summing up over all G_i gives the bound

$$\sum_{G_i} O(\gamma(\kappa_i, \mu_i, c_i)\ell_i) \leqslant O\left(\gamma(\kappa, \mu, c)\sum_{G_i} \ell_i\right) \leqslant O(|V| 2^{(d-1)\kappa}\gamma(\kappa, \mu, c)).$$

All together, we get a total time of

$$O(|V|^{c+1}((d-1)\kappa + \mu)^{\kappa+2}(d+1)^{\mu+d\kappa}2^{d\kappa}). \qquad \square$$

Again the evaluation of the Δ-function can also be done in parallel with a logarithmic number of iterations, which gives Theorem 9.

10. Conclusions

We have shown that the single-source broadcasting problem remains hard for planar networks of bounded degree if the internal connectivity is high, that means there is no edge- or node-decomposition with components of small size. On the other hand, even a much more general version with many sinks and individual deadlines can be solved efficiently on graphs that can be decomposed nicely.

Thus, one can conclude that generating optimal broadcast schedules is a difficult task in general. The intuition that this must be due to a complex structure of the network which gives a lot of freedom designing a schedule has been verified by a rigorous proof. Most interestingly, such structures can already occur in bounded degree planar graphs.

References

[1] S. Arnborg, J. Lagergren, D. Seese, Easy problems for tree-decomposable graphs, J. Algorithms 12 (1991) 308–340.

[2] J.-C. Bermond, P. Hell, A. Liestman, J. Peters, Broadcasting in bounded degree graphs, SIAM J. Discrete Math. 5 (1992) 10–24.

[3] H. Bodlaender, T. Kloks, Better algorithms for the pathwidth and treewidth of graphs, Proceedings 18'ICALP, 1991, pp. 544–555.
[4] G. Buntrock, A nice special case of 3DM is $\mathcal{NP}$-complete, Technical Report, TU Berlin, 1984.
[5] M. Dyer, A. Frieze, Planar 3DM is $\mathcal{NP}$-complete, J. Algorithms 7 (1986) 174–184.
[6] M. Garey, D. Johnson, Computers and Intractability, A guide to the theory of $\mathcal{NP}$-completeness, Freeman, San Fransisco, 1979.
[7] S. Hedetniemi, S. Hedetniemi, A. Liestman, A survey of gossiping and broadcasting in communication networks, Networks 18 (1988) 319–349.
[8] W. Hohberg, The decomposition of graphs into k-connected components for arbitrary k, Technical Report, TH Darmstadt, 1990.
[9] W. Hohberg, R. Reischuk, Decomposition of graphs – a uniform approach for the design of fast sequential and parallel algorithms on graphs, Technical Report, TH Darmstadt, 1989.
[10] J. Hromkovič, C.-D. Jeschke, B. Monien, Optimal algorithms for dissemination of information in some interconnection networks, Proceedings 15th MFCS, 1990, pp. 337–346.
[11] A. Jakoby, R. Reischuk, C. Schindelhauer, The complexity of broadcasting in planar and decomposable graphs, Technical Report A-95-08, Med. Universität zu Lübeck, 1995.
[12] J. Lagergren, Efficient parallel algorithms for tree-decomposition and related problems, Proceedings 31st FoCS, 1990, pp. 173–182.
[13] A. Liestman, J. Peters, Broadcast networks of bounded degree, SIAM J. Discrete Math. 4 (1988) 531–540.
[14] M. Middendorf, Minimum broadcast time is $\mathcal{NP}$-complete for 3-regular planar graphs and deadline 2, Technical Report, Universität Karlsruhe, 1992.
[15] R. Reischuk, An algebraic divide-and-conquer approach to design highly parallel solution strategies for optimization problems on graphs, Technical Report, TH Darmstadt, 1991.
[16] R. Reischuk, Graph theoretical methods for the design of parallel algorithms, Proceedings 8th FCT, 1991, pp. 61–67.
[17] N. Robertson, P. Seymour, Graph minors II. Algorithmic aspects of tree-width, J. Algebra 7 (1986) 309–322.
[18] P. Slater, E. Cockayne, S. Hedetniemi, Information dissemination in trees, SIAM J. Comput. 10 (1981) 692–701.